Reciprocity without losing causality

A first research result on simplifying finite feedback systems

Proofs and reproducible finite checks completed.
Priority and external mathematical review remain open.

The question

When parts of a system affect one another, can we simplify each part and still preserve what happens when we intervene? Matching each component's input-output rule turns out to be insufficient: a simplification can invent a stable state that the original system cannot occupy.

The sharp rule

Assume the fine system has exactly one solution after every point intervention, including no intervention. A point intervention fixes chosen components and leaves the other equations in force. Assume also that the component rules commute exactly with the chosen simplifications.

Classify each component as **fully retained**, **wholly erased**, or **partially simplified**. The third kind keeps some distinctions while merging others.

Every compatible fine system on a given dependency graph is guaranteed to retain its unique interventional solutions if and only if there is no directed cycle entirely among the partially simplified components.

This is a universal guarantee from the graph and the interface. An individual system may still be safe when this graph test fails. The whole graph can contain feedback cycles; the theorem does not require an acyclic fine system.

The smallest failure

Two components each have values 0, 1 and 2. The first copies the second; the second applies a swap that fixes 0 and exchanges 1 with 2. Their only equilibrium is (0, 0). Fixing either component always determines the other uniquely.

FINE SYSTEM: FOUR DIFFERENT STATES KEEP CHANGING



COARSE DESCRIPTION: ALL FOUR LOOK LIKE (ON, ON)

An unchanging description need not describe an equilibrium.

Now label 0 as OFF and merge 1 and 2 into ON. Each coarse rule copies the other coarse value, so both (OFF, OFF) and (ON, ON) appear to be equilibria. The second has no fine equilibrium behind it. Two three-state components are minimal in the stated loopless class; all minimum failures are this example up to relabelling.

Proof, checks and repair

Why the rule works

Take any proposed coarse equilibrium after an intervention. Temporarily fix all fully retained components to the fine values it specifies. The fine system's intervention guarantee supplies one solution for the remaining components. Wholly erased components have no coarse distinction to get wrong. The partially simplified components form an acyclic subsystem, so their coarse values are forced in order. The temporary fixes can then be removed: injectivity ensures their original equations hold. Every coarse equilibrium therefore comes from the unique fine equilibrium.

Conversely, a cycle of partially simplified components can hide a binary distinction at each component. Copy those hidden distinctions around the cycle with one reversal. The fine equations cannot close in that region, while merging the distinctions makes them appear to agree. Add a separate fixed state to obtain a fine system with unique solutions under every intervention. This constructs the obstruction on every forbidden cycle.

Exact finite verification

Search	Coverage	Result
Two components	900 systems; all partitions; 5,206 simple compatible pairs	18 failures, all minimal and equivalent by relabelling
Three Boolean components	4,096 systems; all 8 identity/erasure interfaces	2,928 simple compatible pairs, all safe
Sharpness witnesses	64 graphs x 27 interface-type assignments	135 forbidden cases; 135 verified witnesses
Retention refinements	23,276 upgrades of 7,476 compatible pairs	Exact descent criterion agrees in every case

Two separately written enumerators agree on their overlapping searches, including every two-component domain subtotal. Independent internal AI reviews found no mathematical defect. These checks corroborate the proofs; they are not a proof assistant verification or external peer review.

Retaining detail has dependencies

A second proposition gives an exact repair rule for upgrades that retain a whole component. If an upgraded output distinguishes two input values that the interface still merges, that input component must also be retained. Repeating this rule gives a closure under hidden-input dependencies. A valid certificate upgrade must both satisfy this closure and break every cycle among the remaining partial components.

In the smallest example, retaining one component breaks the graph cycle but makes its mechanism ill-defined at the remaining coarse input. Both components must be retained. The exact optimization counts the union of all forced upgrades, avoiding double-counted dependencies.

Essential boundary

A unique solution without intervention is not enough. Nor does unique solvability under every intervention imply that synchronous or fair asynchronous updates converge. Separate explicit counterexamples and checks are included in the research package.

Why Hegel; why Hegelese

A philosophical constraint, made testable

The project starts from the Hegel Reader's discussion of formal causality (Day 93) and the universal concept (Day 97). The primary anchor is Hegel's Encyclopaedia, sections 154-159. Section 156 argues that merely saying two sides cause one another does not yet explain their mediation within the whole.

Our mathematical interpretation is limited and explicit: a description of interacting parts should preserve their joint causal possibilities, including the effects of intervention. The false equilibrium example shows how a higher-level unity can be produced merely by hiding a fine-level failure to agree. The theorem states when this cannot happen within the chosen finite model.

Feedback is not thereby identified with Hegelian freedom, and a fixed point is not the whole Concept. Unique solvability is a modeling assumption, not something deduced from Hegel. Some Reader chapter labels and analogies are unreliable; the interpretation was checked against the original text.

A concrete prospect for Hegelese

Hegelese already checks specified revisions of finite processes. A research extension could ask a further question: does this simplification remain causally faithful when the parts are connected in feedback? An experimental certificate would record the component maps, mechanism equations, a separate justification of fine simplicity, and the acyclic graph on partial components.

A failed certificate would identify a cycle or a hidden-input distinction that must be retained. It would distinguish a missing structural guarantee from an actual demonstrated semantic failure. This package proposes that interface; it does not change the language or claim that the existing checker implements it.

What may be new

Cyclic causal abstraction, value merging, and the collapse of hidden cycles into apparent fixed points are already studied. The strongest candidate here is the exact universal classification by cycles among maps that are neither injective nor constant, together with its minimal obstruction. The targeted literature audit found no matching theorem. That is a reason for further review, not proof of priority.

Selected sources and next question

Hegel Reader: Day 93 and Day 97. Hegel, Encyclopaedia, sections 154-159.

Lorenz and Tull (2026), Causal and Compositional Abstraction, sections 6 and 8. Bongers et al. (2021), Foundations of Structural Causal Models with Cycles and Latent Variables, sections 5 and 8.

Geiger et al. (2025), Causal Abstraction, Definitions 25 and 38. Steggles (2011), Abstracting Asynchronous Multi-Valued Networks, Definition 15.

Next: characterize safe abstractions for fixed installed mechanisms when the universal graph test fails, and optimize genuinely partial refinements rather than retaining whole components.

Companion files: theorem.md; minimal-counterexample.md; retention.md; PRIOR-WORK.md; PROOF-REVIEW.md; enumerate_models.py; independent_check.py; exact JSON reports. The full proofs and verification scope are in those files.